The Role of Game Theory in Information Warfare[†]

Samuel N. Hamilton, Wendy L. Miller, and Allen Ott
ORINCON Information Assurance
9363 Towne Centre Drive, San Diego, CA 92121-3017

O. Sami Saydjari
SRI Computer Science Laboratory
3601 43rd Street South, Wisconsin Rapids, WI 54494

Protection of cyber assets is critical in today's corporate and military environment. Whether an attacker is a casual hacker or an organized terrorist group, it is crucial to be able to keep your system functional and secure. Game theory offers an array of promising techniques for aiding tactical analysis in this domain. In this paper, we identify the areas of game theory relevant to information warfare, and present an example of these techniques in action.

1 Introduction

Currently, information defense relies to a large extent on human review of low level system data to address short term security concerns. While this is sufficient for repelling casual hackers using well known techniques, there is a critical need in both the military and in industry for tools to defend against patient, well organized attackers. By framing the attack and defense of critical cyber assets as a game of moves and counter moves, we can utilize well developed game theory algorithms to help predict future attacks and suggest courses of action (COA) that can defend against them with a minimum cost.

The advantages gained through utilization of game theoretic techniques are numerous. First, game theory provides the capability of examining hundreds of thousands of possible scenarios. This allows computer analysis to provide a vastly different, yet extremely valuable, perspective from human experts, who concentrate their efforts on what they view as the top few most dangerous possibilities. Second, game theory can provide methods for suggesting several potentials courses of action with accompanying predicted outcomes. This is a significant improvement on COA generation techniques that only provide one suggestion, and cannot elucidate the motivating factors behind it. Third, game theoretic techniques have proven quite effective at analyzing what-if scenarios, allowing detailed analysis of important chains of events, and utilizing that analysis at a later date.

In this paper, we outline the areas of game theory of particular relevance to the domain of information warfare. The paper is organized as follows: in section 2 we present an example scenario illustrating the role we foresee game theoretic techniques taking in an information warfare setting. Section 3 summarizes the current state of relevant game theoretic techniques, and in section 4 we draw conclusions regarding their potential in information warfare.

2 Motivating Example

The conclusions and actions of the regional commander's Centralized Defense Controller (CDC) in the following attack scenario illustrate the role we foresee game theory tools and techniques fulfilling. For this example we chose a simple, generic network of cyber assets. The game theory component resides in the CDC, which is responsible for defending four enclaves (A-D). Three (A,B,C) have their own router and firewall providing access to the internet. D is a completely separate network, but is occasionally hooked up to C for internet access. The public has physical access to Network A during the day; network B requires challenge-response authentication at all times; and networks C&D require strong certificate-based authentication. Sensitive battle planning is done on D; planning support like weather on C; routine data such as payroll on B; public relations and access, such as web presence on A. Our system goals are as follows:

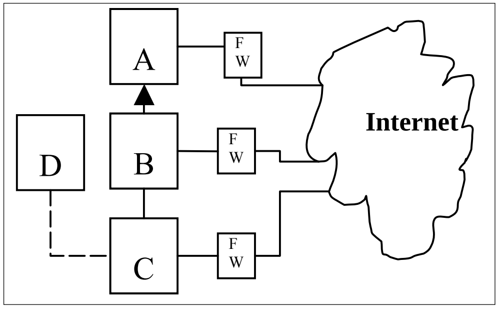

Figure 1. example network

[†] This material is based on work sponsored by DARPA and supported by the Air Force Research Laboratory under Contract F30602-00-C-0165.

1. Prevent compromise, corruption, or denial of service of planning and support data or systems; 2. Accurately and quickly detect, react, and respond to intrusions.

Scenario: Observed tension in region X, entered into a situation database, triggers CDC to respond by increasing monitoring activity on related networks and restricting non-essential access from and to external systems. The monitoring reveals sub-threshold port scans indicative of an intelligence-gathering attack phase. Based on CDC's cyber war-gaming knowledge base of "opening moves", the system predicts attempts to more deeply map the CDC's networks and targeting of the command and control system because of adversary's interest in planning information. In response, CDC isolates network D and runs daily scans for program changes. The CDC cyber-wargaming subsystem then predicts that the attacker will attempt to gain a leverage point by attempting to take over network A because it continues to be a public access point. Based on these predictions, the system creates a fishbowl and shunts all external traffic for network A to the fishbowl. The fishbowl is designed to send deceptive information about system internals to the attacker and attempt to fully capture his actions for further analysis of intent. The CDC predicts that the attacker will attempt to corrupt the plan-supporting systems on network C, or, failing that, launch a distributed denial of service attack against B,C, and D to prevent the kinetic warfare planning process from continuing. In response, the CDC creates a redundant computing site to B, C, and D and mirrors data so that planning can continue if B, C, and D are disabled. A few weeks later, analysis detects corruption of system files by unknown means; life-cycle attacks are suspected. Because of the complexity of the situation, the CDC does not simply respond automatically, it presents several strategic COA options to the decision-maker. P1, the conservative option, assumes the adversary compromised the kinetic plan based on a supposition of multiple points of infiltration, only one of which having been detected. The recommended response is to throw out the kinetic plan, abandon the current systems for planning, and move the operation to the backup site. This will take 3 days. P2, the middle-ground option, assumes that the detected attacker insert is the only one and that the attacker's insight into the system only goes back to the last scan of the system. The recommended response is to restore the affected system file, reconfigure the system resources in case the attacker was able to exfiltrate inside information, rollback data to the point of the last scan, and increase the system scan rate to hourly. This will take 24 hours. P3, the risky option, is to assume no data was lost or corrupted, leave the sniffer in the system, fishbowl it. and continue operations normally, with further increased monitoring, including the hourly scan rate. This will take 1 hour.

In the scenario, first notice how input used to improve situational awareness is taken from a variety of sources: human, network data, previous events, and even scans that the system initiates. This data is used to consider thousands of scenario variations representing a series of moves by us and our opponents. The computer can then determine the likelihood, method, and cost of these scenarios. Second, notice that even if the system predicts the adversary's next move incorrectly, its response can still be effective by thwarting several potential attacks simultaneously, such as the data corruption and denial of service attacks in the above scenario. One can imagine that with unlimited time and under no stress, humans could develop similar responses. Unfortunately, ideal conditions are not the norm – especially during conflicts. People make mistakes, as well as overlook possibilities and lessons learned from the past. Further, it is often difficult for people to consider move-counter-move sequences deeply, so it is difficult to avoid starting down a move sequence that is ultimately disadvantageous. This is where the value of game theory comes in. By considering huge numbers of possibilities, computers have been shown capable of finding exceptions to general rules in numerous games such as chess and backgammon, and exploiting them mercilessly. In some cases, whole new theories on how to play are developed because of this [1]. Our goal is not to replace the human analyst, but to supply him with a powerful tool for suggesting approaches and techniques that might not occur to him, and provide the ability for better and more detailed analysis.

3 Game Theory

In this section we describe the current state of game theory, concentrating on areas relevant to tactical analysis in information warfare. There are two fundamental pieces to a tactical analysis engine: the search technique, and the evaluation function. The evaluation function reports how good a position looks to the player whose move it is.

The search technique determines which moves to consider, and returns the move (or COA) it considers best at the end of the search. The search forms a tree of moves and counter moves, and the leaves are judged using an evaluation function. The most popular technique is to use a derivative of mini-max [2], which propagates leaves back to the root by having parents alternatively assigned the minimum/maximum value of its children, ending with the root maximizing. This works under the assumption that the opponent evaluation function ε_o is the opposite of your own evaluation function ε_s. Thus, $\varepsilon_o = - \varepsilon_s$.

In the domain of command and control planning, there has been some limited success based on the assumption that $\varepsilon_o = - \varepsilon_s$ [2]. We believe, however, that in the domain of information warfare this assumption is not justified. There are a variety of reasons for $\varepsilon_o \neq - \varepsilon_s$. First, your opponent likely has different goals and priorities than you. Second, they may not have full information about your network configuration. This makes it more difficult for them to judge how close they are to reaching their goals, which is the purpose of ε_o.

Due to these differences, there are three areas of game theory we will concentrate on because of their particular relevance to information warfare. The first is *pruning*, as the difference between our evaluation function and an opponent evaluation function invalidates many of the traditional techniques. The second is opponent modeling, since it is necessary to define the opponent evaluation function. The third is tuning our own evaluation function.

3.1 Pruning

While we cannot use a mini-max derived search, we can use a max-max′ search, where nodes in the tree alternate between passing the maximum child according to the evaluation function of the player whose turn it is. Unfortunately, pruning techniques available in a max-max′ search are much more limited than in mini-max searches, where alpha-beta pruning [3] can mathematically eliminate many possibilities. Moves are eliminated by showing that searching a move cannot change the outcome of the search. Unfortunately, in max-max′ searches, it has been shown that no such pruning method is possible [4].

Since in any sufficiently complex game some type of pruning is necessary to limit the explosion of the search space with each increase in search depth, alternative techniques have been explored. One group of techniques is derived from best-first searches. In a best-first search, a tree of explored possibilities is kept open, and each leaf is judged as to how promising it is. The most promising node is then expanded, and its children added to the queue. In this scenario, the most promising move is defined as the most likely to be in the final predicted move group. The problem with best-first searches is that the memory requirements are explosive, so in a game where a large number of positions must be considered this solution is not feasible.

β-pruning [5] has been explored specifically with respect to max-max′ searches. In β-pruning, the opponent evaluation function and search depth are assumed to be known, and the opponent is assumed to search using a mini-max based strategy. These assumptions allow the use of weaker versions of *alpha-beta* pruning. It is unclear how realistic these assumptions are unless playing against a known computer opponent running on inferior hardware.

Another search technique proposed for max-max′ searches is αβ* [4]. This technique assumes that the opponent uses an evaluation function similar to your own (i.e. it uses the same heuristics to judge a position) but weights them differently. The technique takes into account the fact that the opponent evaluation function is not completely accurate, but assumes it is correct within a certain error range. Pruning is then done for values outside of that range, using techniques derived from the motivating principals of the alpha-beta search.

3.2 Modeling the opponent evaluation function ε_o

Most literature in this area assumes that the opponent uses some type of mini-max search which goes to a fixed depth [4,5]. The problem then becomes determining what that depth is, and what evaluation function is in use.

One approach to identifying depth and evaluation characteristics is to iteratively increase model depth, and use reinforcement techniques to learn which depth best predicts opponent behavior. If the evaluation function is given, this technique works quickly and effectively [5].

For the evaluation function, the usual assumption is that an opponent evaluation function uses a subset of the heuristics in our own evaluation function, with different weights. A hill climbing algorithm can then be used based on a function of the number of correct opponent move predictions taken from a list of previous opponent moves or games. A Linear programming technique using pattern recognition methods was proposed in [6], and used with some success to find optimal weights in a follow up pass after hill climbing in the domain of checkers [5].

3.3 Determining our evaluation function ε_s

Over the years some very effective techniques have been developed for tuning evaluation functions. In [7], a method was proposed for tuning the weights in a list of heuristics in the domain of checkers. This turned out to be good enough to build a World Champion checker program. A world-class backgammon program used neural networks to learn using similar techniques [1,8]. In computer chess, Deep-Blue rose to the top riding an evaluation function that was also automatically tuned [9].

While there are some differences between the techniques used in each case, there are some significant similarities. First, a human comes up with a list of heuristics expected to be correlated with success or failure. Then, a large number of master level games are analyzed at a low depth, and the best move is selected. This move is checked against the actual move played. If the move matches, the heuristic weighting is reinforced.

There has also been some work on automated methods for heuristic generation in the domain of othello [10]. This work was successful in finding valid heuristics, though they did not actually improve the program results since the slow evaluation function was more of an impediment than the improved accuracy could compensate for. A method of linearly combining Boolean feature lists into heuristics and then weighting them has also been explored [11].

4 Conclusions

We believe that there is the potential for game theory to play a significant role in information warfare. The combination of the ability to consider millions of possibilities, model attacker characteristics, and self generate what-if scenarios seems too potent to ignore. In fact, the potential seems so great that it may only be a matter of time before advanced attackers begin to utilize these techniques offensively, putting any system not similarly equipped at a significant disadvantage. That said, we acknowledge that incorporating the techniques presented in section 3 into a working system is non-trivial. The domain of information warfare is significantly different from games such as chess and backgammon, and these variances create a rich research environment for future work. An analysis of how these differences interact with current game theory can be found in [12].

References

[1] G. Tesauro. Temporal Difference Learning and TD-Gammon. *Communications of the ACM*, 38(3):58-68, 1995.

[2] A. Katz and B. Butler, "Game Commander"-Applying an Architecture of Game Theory and Tree Lookahead to the Command and Control Process, Proceedings of the Fifth Annual Conference on AI, Simulation, and Planning (AIS94), Florida, 1994.

[3] P. Winston, *Artificial Intelligence*, Reading Massachusetts: Addison-Wesley, 1992.

[4] D. Carmel and S. Markovitch, Learning and using Opponent Models in Adversary Search, Technical Report CIS9606, 1996.

[5] H. H. L. M. Donkers e.t. al, Implementing β-pruning Opponent-Model Search, Technical Reports in Computer Science., CS 00-05. IKAT, Universiteit Maastricht, Maastricht, The Netherlands, May 2000.

[6] R. O. Duda and P. Hart. Pattern Classification and Scene Analysis. New York: Wiley and Sons, 1973.

[7] A. L. Samuel. Some studies in machine Learning using the Game of Checkers. *IBM J. of Research and Development*, 3(3):211-229, 1959.

[8] G. Tesauro. TD-Gammon, a Self-Teaching Backgammon Program, reaches master-level play. *Neural Computation*, 6(2):215-219, 1994.

[9] F. Hsu et.al. Deep Thought. In T.A. Marsland and J. Schaeffer, editors, *Computer, Chess, and Cognition*, p. 55-78. Springer Verlag, 1990.

[10] M. Buro, Statistical Feature Combination for the Evaluation of Game Positions, JAIR 3, p. 373-382, 1995.

[11] P. E. Utgoff. Constructive Function Approximation. Technical Report 97-4, Univ. of Mass, 1997.

[12] S. N. Hamilton, W. L. Miller, A. Ott, and O. S. Saydjari, Challenges in Applying Game Theory to the Domain of Information Warfare , *The Information Survivability Workshop*, 2001, submitted.